IN THE CLAIMS

Please cancel Claims 8-17, 19 and 25 without prejudice, amend Claims 1, 20, 26, 32, 34, 35, 40, 42, and 43, and add new Claims 46-50 as follows:

1. (Currently amended) A method of optimizing the instruction set of a digital processor having a mixed 16-bit and 32-bit instruction set architecture, comprising:

(i) providing a program having a plurality of different instruction types, ~~including~~ individual ones of said instruction types being included within at least one of ~~both~~ 16-bit and 32-bit ~~types~~ instruction words;

(ii) determining the static frequency of each of said instruction types from a base instruction set;

(iii) determining the number and type of instructions necessary for correct instruction set execution based at least in part on said act of determining the static frequency; and

(iv) creating a compressed instruction set encoding to generate a compressed instruction set based at least in part on said act of determining;

wherein said act of creating a compressed instruction set comprises creating at least one compressed 14-bit instruction disposed within one of said 16-bit or 32 bit words.

2. (Original) The method of Claim 1, further comprising:

re-evaluating said compressed instruction set using at least said steps (i), (ii), and (iii); and

generating an instruction set encoding for said compressed instruction set.

3. (Original) The method of Claim 1, wherein the act of providing a program comprises providing an assembly language program.

4. (Original) The method of Claim 3, further comprising sorting said instruction types by frequency of usage.

5. (Original) The method of Claim 4, wherein said digital processor includes an extension logic unit adapted to execute at least one extension instruction, and the act

of providing comprises providing a program having said at least one extension instruction, said at least one execution instruction being executable by said extension logic unit.

6. (Original) The method of Claim 1, wherein the act of creating a compressed instruction set comprises selecting "N" instructions having the greatest frequency of occurrence, said selected "N" instructions permitting said program to be compiled with a predetermined size.

7. (Original) The method of Claim 6, further comprising the act of determining a compression ratio for said compressed instruction set, said compression ratio being related to the ratio of the number of compressed instructions to the total number of original instructions.

8. – 19. (Cancelled)

20. (Currently amended) An application specific integrated circuit (ASIC), comprising:

a first processor core, said processor core having a pipeline with at least instruction fetch, decode, and execute stages associated therewith;

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set having a predetermined number and type of instructions, said predetermined number and type based at least in part on the static frequency of occurrence of instructions within said base instruction set, said optimized instruction set further comprising at least one extension instruction adapted to perform at least one specific operation;

at least one storage device adapted to store a plurality of data bytes therein, said at least one storage device being accessible by said first processor core; and

at least one extension logic unit adapted to facilitate execution of said at least one execution instruction;

wherein said processor core is adapted to execute both 16-bit and 32-bit instructions without processor mode switching; and

further wherein said pipeline is operative to utilize said optimized instruction set without instruction translation.

21.(Original) The ASIC of Claim 20, further comprising a second processor core, said second processor core being disposed on the same die as said first processor core.

22.(Original) The ASIC of Claim 21, wherein said second processor core comprises a digital signal processor (DSP), said DSP being adapted to perform at least one operation on data provided to said ASIC.

23. (Original) The ASIC of Claim 22, wherein said DSP is adapted for initiation by an instruction from said first processor core.

24. (Original) The ASIC of Claim 22, wherein at least a portion of the operation of said DSP is controlled by extension registers associated with said first processor core.

25. (Cancelled)

26. (Currently amended) A method of operating an extended pipelined digital processor having an instruction pipeline comprising at least instruction fetch, decode, and execute stages, a storage device configured to hold a plurality of program instructions, and an optimized instruction set, the method comprising:

providing a base instruction set having a plurality of 16-bit and 32-bit instructions;

providing an extension instruction set determined at least in part by selections made by a user;

providing a compressed instruction set derived at least in part from said base and extension instruction sets;

assigning one of a plurality of predetermined values to at least one bit within a status register within said processor;

executing at least one instruction from said base instruction set within said pipeline based on a first predetermined value present in said status register; and

executing at least one instruction from said compressed instruction set within said pipeline based on a second predetermined value present in said status register[[.]] ;

<u>wherein 16-bit instructions from said plurality of 16-bit and 32-bit instructions are processed as 16-bit instructions, and 32-bit instructions from said plurality of 16-bit and 32-bit instructions are processed as 32-bit instructions.</u>

27. (Original) The method of Claim 26, wherein the act of assigning comprises assigning a "1" or "0" value to a low address (L) bit within said register.

28. (Original) The method of Claim 27, wherein the act of providing a compressed instruction set comprises encoding at least a portion of the instruction operation codes (opcodes) for the compressed instructions within a predetermined number of the most significant bits of an instruction word.

29. (Original) The method of Claim 27, wherein the act of providing a compressed instruction set further comprises encoding a plurality of said compressed instructions with source register fields located in a predetermined relationship to one another within said compressed instructions.

30. (Original) The method of Claim 29, wherein the act of encoding with said predetermined relationship comprises encoding the source register fields for respective ones of said plurality of compressed instructions at identical locations.

31. (Original) The method of Claim 26, wherein the act of providing a compressed instruction set comprises encoding all of the immediate data fields such that they start from the least significant bit (LSB).

32. (Currently Amended) A user-configured and extended pipelined RISC processor, comprising:

a processor core having an instruction pipeline comprising at least instruction fetch, decode, and execute stages;

a data interface in data communication with said processor core, said interface adapted for data communication with a storage device configured to hold a plurality of program instructions; and

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set being generated by the method comprising:

determining the static frequency of each of ~~said~~ a plurality of instruction types from said base instruction set;

determining the number and type of instructions necessary for instruction set execution based at least in part on said act of determining the static frequency; and

creating a compressed instruction set encoding to generate said compressed instruction set, said creating comprising using an encoding structure having an opcode and two instruction slots, each of said slots having two 14-bit instructions, and selecting two instructions having the greatest frequency of occurrence, said selected two instructions permitting said program to be compiled with a predetermined size;

wherein any immediate data fields in said compressed instruction set start from the least significant bit of respective ones of said instructions; and

wherein a plurality of instructions from said compressed instruction set each have at least one source register field located in common location therein.

33. (Previously presented) The processor of Claim 32, wherein said structure comprises a 32-bit encoding structure with said opcode disposed within the last four bits thereof.

34. (Currently amended) A user-configured and extended pipelined RISC processor, comprising:

a processor core having an instruction pipeline comprising at least instruction fetch, decode, and execute stages;

a data interface in data communication with said processor core, said interface adapted for data communication with a storage device configured to hold a plurality of program instructions; and

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set being generated by the method comprising:

determining the static frequency of each of said instruction types from said base instruction set;

determining the number and type of instructions necessary for instruction set execution based at least in part on said act of determining the static frequency; and

creating a compressed instruction set encoding to generate said compressed instruction set, said creating comprising using a 32-bit encoding structure having an opcode and a plurality of instruction slots, said opcode being disposed within at least the last 4-bits of said structure, and selecting a corresponding number of instructions having the greatest frequency of occurrence, said selected instructions permitting said program to be compiled with a predetermined size;

wherein both said user-extension and configuration of said processor are performed as part of generating of a description language model of said processor; and

wherein only a subset of available registers, and implied registers, are used by said compressed instruction set, said subset and implied registers reducing the number of bits required to encode a register.

35. (Currently amended) A method of operating a user-extended and configured RISC processor having an instruction pipeline comprising at least instruction fetch, decode, and execute stages, a storage device configured to hold a plurality of program instructions, and an optimized instruction set, the method comprising:

providing a base instruction set having a plurality of instructions and an extension instruction set having at least one extension instruction;

providing a compressed instruction set derived at least in part from said base instruction set;

assigning a value to a compressed instruction selection bit within a status register within said processor;

executing at least one instruction from said base instruction set within said pipeline based on a first predetermined value present in said status register; and

executing at least one instruction from said compressed instruction set within said pipeline based on said assigned value of said compressed instruction selection bit in said status register, wherein instructions from said set of compressed instructions are executed directly without translation.

36. (Previously presented) The method of Claim 35, wherein the act of providing a compressed instruction set comprises encoding at least a portion of the instruction operation codes (opcodes) for the compressed instructions within a predetermined number of the most significant bits of an instruction word.

37. (Previously presented) The method of Claim 36, wherein the act of providing a compressed instruction set further comprises encoding a plurality of said compressed instructions with source register fields located in a predetermined relationship to one another.

38. (Previously presented) The method of Claim 37, wherein the act of encoding with said predetermined relationship comprises encoding the source register fields for respective ones of said plurality of compressed instructions at identical locations.

39. (Previously presented) The method of Claim 35, wherein the act of providing a compressed instruction set comprises encoding all of the immediate data fields such that they start from the least significant bit (LSB).

40. (Currently Amended) A method of operating an extended pipelined digital RISC processor having an instruction pipeline comprising at least instruction fetch, decode, and execute stages, a storage device configured to hold a plurality of program instructions, and an optimized instruction set comprising both 16-bit and 32-bit instructions, the method comprising:

providing an extension instruction set having a plurality of user-selected extension instructions;

providing a compressed instruction set derived at least in part from said extension instruction set;

encoding a plurality of compressed instructions from said compressed instruction set into an instruction word having an op-code;

assigning one of a plurality of predetermined values to at least one bit within a status register within said processor; and

executing at least one of said compressed instructions from said instruction word within said pipeline based on a second predetermined value present in said status register;

wherein a majority of instruction opcodes are encoded within the top n bits, n being an integer greater than 1, said top n bits determining the format of the remaining bits within an instruction associated therewith.

41. (Previously presented) The method of Claim 40, wherein said act of encoding comprises encoding two 14-bit compressed instructions into a 32-bit aligned instruction word having said opcode disposed within at least the last four bits thereof.

42. (Currently Amended) A reduced instruction ~~rest~~ set (RISC) pipelined digital processor, comprising:

a user-configured and user-extended RISC processor core having an instruction pipeline comprising at least instruction fetch, decode, and execute stages;

a data interface in data communication with said processor core, said interface adapted for data communication with a storage device configured to hold a plurality of program instructions; and

an optimized instruction set comprising a base instruction set and a compressed instruction set, said compressed instruction set being generated by the method comprising:

determining the static frequency of each of said instruction types from said base instruction set;

determining the number and type of instructions necessary for instruction set execution based at least in part on said act of determining the static frequency; and

creating a compressed instruction set encoding to generate said compressed instruction set; and

wherein both said user-extension and configuration of said processor core are performed as part of generating of a description language model of said processor core; and

wherein instructions from said compressed instruction set are processed in the same form as they are stored in said storage device.

43. (Currently Amended) The processor of Claim 42, wherein said generating of a description language model of said processor core comprises:

receiving one or more inputs from a user for at least one customized parameter of the processor core; and

generating through an automated process said description language model based at least in part on the at least one customized parameter, and at least one prototype description, and at least one extension logic description.

44. (Previously presented) The processor of Claim 43, wherein said generating of a description language model of said processor core further comprises generating, via an automated process, test code associated with said model.

45. (Previously presented) The processor of Claim 43, wherein the description language model includes both functional and structural description language descriptions for the processor core.

46. (New) The method of Claim 1, wherein said at least one compressed 14-bit instruction comprises a register-register multi-use opcode14-bit instruction encoding comprising a 5-bit op-code, and two 3-bit fields.

47. (New) The method of Claim 1, wherein said at least one compressed 14-bit instruction comprises a compressed paired-instruction op-code, and two compressed 14-bit instructions.

48. (New) A RISC microprocessor having a processor core and at least one instruction set operative to run on said core, said at least one instruction set being defined by a plurality of instructions comprising:

a multi-bit offset, 14-bit conditional branch instruction;

a register-register multi-use opcode 14-bit instruction;

a single source register opcode 14-bit instruction;

an implied register opcode 14-bit instruction;

a multi-bit offset, 14-bit Load instruction;

a multi-bit offset, 14-bit Store instruction;

a multi-bit offset, 14-bit LOAD/STORE instruction;

a multi-bit integer, 14-bit MOVE/COMPARE instruction; and

a multi-bit integer, 14-bit ADD instruction.

49. (New) A RISC microprocessor having a processor core and at least one instruction set operative to run on said core, said at least one instruction set being defined by a plurality of instructions comprising:

a multi-bit signed offset conditional branch, 15-bit instruction;

a multi-bit register-register multi-use opcode instruction;

a multi-bit single source register opcode instruction;

a multi-bit implied opcode instruction;

a multi-bit offset, 15-bit LOAD instruction;

a multi-bit offset, 15-bit STORE instruction;

a multi-bit offset, 15-bit LOAD/ STORE instruction; and

a multi-bit integer, 15-bit MOV/ CMP instruction.

50. (New) The microprocessor of Claim 49, wherein said plurality of instructions further comprises:

a 5-bit integer, 15-bit instruction;

a multi-bit MOV.F instruction;

a multi-bit integer, 15-bit ADD/SUB instruction;

a multi-bit integer, 15-bit LOAD instruction;

a multi-bit integer, 15-bit ADD instruction;

a multi-bit signed offset branch-and-link, 15-bit instruction; and

a 15-bit instruction reserved for future expansion.